

WSTĘP DO PROGRAMOWANIA W *LABVIEW* – cz. I

5.1. CEL I ZAKRES ĆWICZENIA

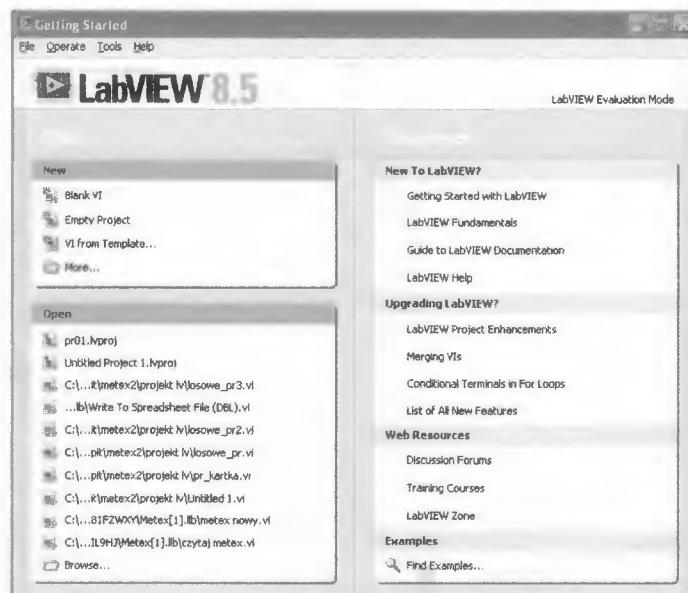
Zapoznanie się studentów z podstawami obsługi programowania w języku graficznym LabView służącym do programowania systemów pomiarowych.

5.2. WPROWADZENIE DO *LABVIEW*

5.2.1. Wstęp.

LabVIEW jest zintegrowanym środowiskiem programowania opracowanym (i sukcesywnie usprawnianym) przez firmę National Instruments. Jest to pakiet oprogramowania służący do projektowania wirtualnych przyrządów pomiarowych. Obecnie najnowszą dostępną wersją pakietu jest wersja 7.0. Główne cechy środowiska LabVIEW to:

- obiektowe podejście do programowania (każdy element płyty czołowej, funkcji wewnętrznej programu lub podprzyrządu traktowany jest jak obiekt z właściwościami i metodami z nim związanymi),
- programowanie odbywa się w języku graficznym „G”, który w przeciwieństwie do klasycznych języków oferuje ikony funkcji łączone na diagramach przy pomocy „nitek” zamiast formy tekstowej
- wykorzystywanie bibliotek dołączanych dynamicznie (DLL) umożliwia korzystanie z funkcji standardowo niedostępnych w pakiecie LabVIEW. Jest to przydatne podczas tworzenia skomplikowanego lub nietypowego przyrządu wirtualnego
- łączność ze specjalizowanymi urządzeniami takimi jak karta akwizycji danych odbywa się przy pomocy standardu bibliotecznego wejścia/wyjścia VISA. Jest to standard uwzględniający wszystkie najważniejsze interfejsy wykorzystywane w metrologii, zapewniający uniwersalność pisanych aplikacji
- gotowy program może zostać skompilowany do postaci samodzielnego programu dzięki jednemu z wewnętrznych narzędzi pakietu. Dzięki temu może on być uruchamiany na komputerach bez zainstalowanego środowiska LabVIEW, jednak z pewnymi obostrzeniami (wymagana jest instalacja na stanowisku docelowym środowiska uruchomieniowego LabVIEW Run-Time).



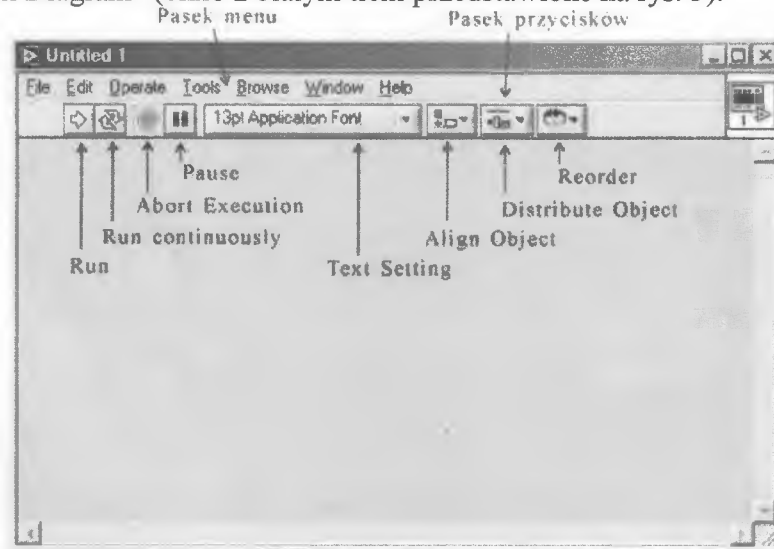
Rys. 1. Okno startowe LabView.

Przedstawione cechy czynią z LabVIEW środowisko niezwykle popularne, dostępne na wielu platformach programowych i sprzętowych (poczynając od Windows, poprzez UNIX-a, a kończąc na komputerach typu Macintosh). Pakiet jest stosunkowo łatwy w obsłudze, a intuicyjny sposób tworzenia kodu sprawia, iż jest on w obrębie zainteresowań nie tylko inżynierów, ale również osób na co dzień nie związanych z projektowaniem przyrządów.

5.2.2. Uruchomienie programu LabView.

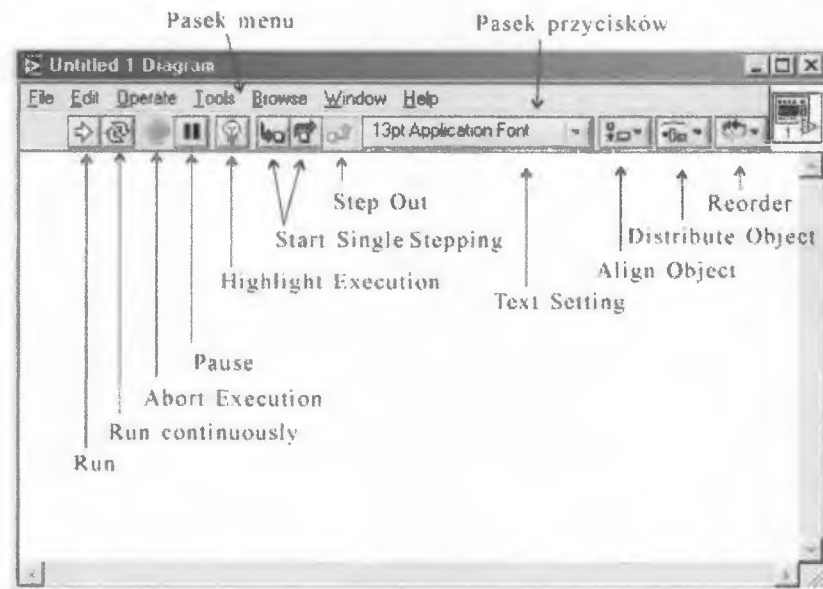
Po pojawieniu się okna przedstawionego na rys. 1 wybrać opcję "BlankVI" VI"..

Na ekranie pojawią się dwa okna podstawowe: "Front Panel" (okno z szarym tłem przedstawione na rys.2) oraz "Block Diagram" (okno z białym tłem przedstawione na rys. 3).



Rys.2. Okno podstawowe: "Front Panel".

Okno "Front Panel" stanowi interfejs użytkownika korzystającego z programu napisanego w LabView. Przy jego pomocy tworzymy panel, na którym możemy umieścić przyciski, przełączniki, wskaźniki, okna dialogowe, okna z wynikami, wykresy. Przy pomocy tego okna użytkownik programu steruje wykonaniem programu i odczytuje wyniki wykonania programu.

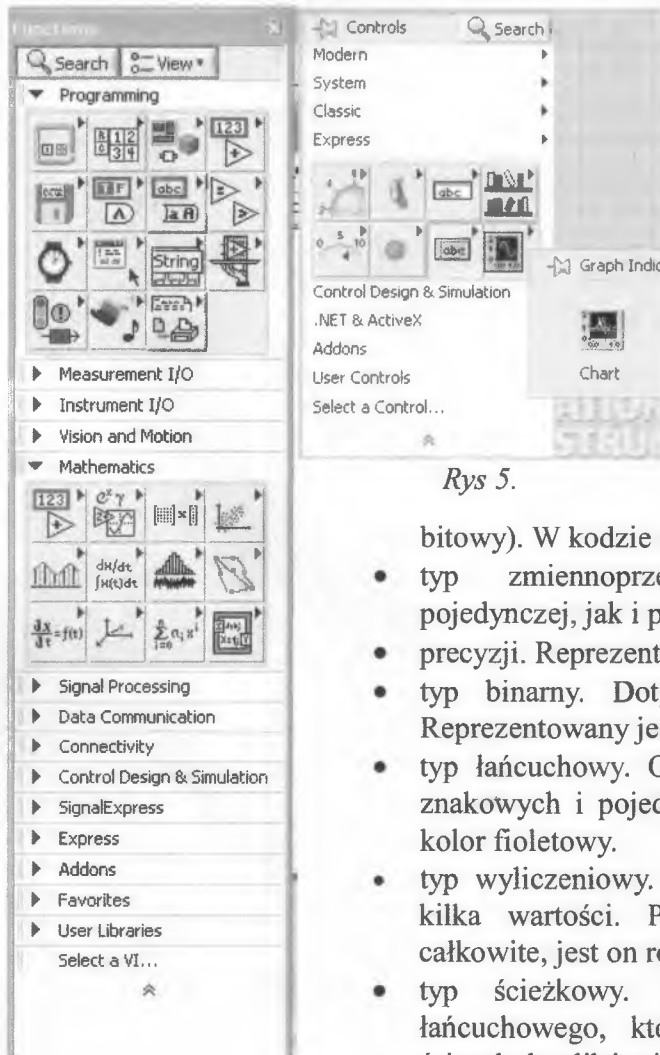


Rys. 3. Okno podstawowe "Block Diagram".

Okno "Block Diagram" zawiera kod programu napisany w języku graficznym. Wszystkie instrukcje języka graficznego LabView posiadają swoją reprezentację graficzną (obiekty graficzne). Programowanie polega na wyborze odpowiednich instrukcji i ich łączeniu zgodnie z kolejnością przepływu danych lub kolejnością wykonywania obliczeń.

Na rys. 4 przedstawiono zestaw narzędzi „Functions”. Zestaw ten dostępny jest dla okna podstawowego „Block Diagram”.

Na rys. 5 pokazano przykład okna „Controls” dostępny dla okna „FrontPanel”.



Rys 5.

5.2.3. Typy danych

LabVIEW obsługuje większość typów danych właściwych innym językom programowania. W przeciwieństwie do nich, tutaj deklaracja typu reprezentowana jest przez kolor obiektu lub nitki łączącej. Generalnie typy danych można podzielić na proste i złożone.

1. Podstawowe typy proste to:

- typ całkowitoliczbowy. Obejmuje on zarówno typy ze znakiem (integer 8-, 16- i 32-bitowy), jak i bez znaku (unsigned integer 8-, 16- i 32-bitowy). W kodzie reprezentowany jest on przez kolor niebieski.
- typ zmiennoprzecinkowy. Obejmuje zarówno liczby pojedynczej, jak i podwójnej
- precyzji. Reprezentowany jest przez kolor pomarańczowy..
- typ binarny. Dotyczy obiektów o logice dwuwartościowej. Reprezentowany jest przez kolor zielony.
- typ łańcuchowy. Odpowiada za wszystkie operacje na łańcuch znakowych i pojedynczych znakach. Reprezentowany jest przez kolor fioletowy.
- typ wyliczeniowy. Dotyczy obiektów mogących posiadać tylko kilka wartości. Ponieważ typowi temu odpowiadają liczby całkowite, jest on również reprezentowany przez kolor niebieski.
- typ ścieżkowy. Właściwa dla LabVIEW odmiana typu łańcuchowego, która służy niemal wyłącznie do podawania ścieżek do plików i katalogów.
- typ identyfikatora sesji VISA. Odmiana typu łańcuchowego służąca do obsługi funkcji odpowiedzialnych za sesję VISA.

Rys. 4.

Reprezentuje ją kolor bordowy.

- w obrębie większości typów danych istnieje subtelne rozróżnienie pomiędzy typami jedno- i wielowymiarowymi. Te ostatnie mają ten sam kolor, co odpowiadające im typy jednowymiarowe, jednak ich nitki są grubsze. Do obsługi takich danych są oczywiście potrzebne inne kontrolki na panelu czołowym.

2. Typy złożone (inaczej klastry) to odpowiedniki struktur w klasycznych językach programowania. Są to zmienne zawierające w sobie więcej niż jedną zmienną (mogą być one różnych typów). Charakteryzują je grube, prążkowane linie.

- ❖ struktura obsługi błędów, zawierająca trzy najważniejsze komponenty:
 - informacja o wystąpieniu błędu (typ binarny)
 - numer błędu (liczba całkowitoliczbową: dodatnia, jeśli wystąpiło ostrzeżenie, ujemna dla błędu)
 - opis źródła błędu (zmienna łańcuchowa)

- ❖ Struktura obsługi błędów jest reprezentowana przez kolor fioletowy.
 - typ przebiegów sygnałowych (waveform). Jest to rodzaj klastra, który zawiera
 - specyficzne dane dotyczące wektora sygnału. Są to przede wszystkim:
 - to – moment rozpoczęcia generacji sygnału (decyduje o offsecie)
 - dt – odległość na osi czasu między próbkami sygnału (determinowana przez częstotliwość próbkowania)
 - Y – wektor próbek sygnału
- ❖ Typ przebiegów sygnałowych jest reprezentowany przez kolor brązowy

5.2.4. Struktura programu

Najistotniejszymi elementami kodu LabVIEW są funkcje oraz podprzrzędy, które wykonują większość zadań programu. Zasadnicza funkcja między funkcjami a podprzrzędami test taka, że te ostatnie zawierają płytę czołową, przy pomocy której można dokonywać nastaw. Dodatkowo, mogą też działać jako samodzielne programy. W LabVIEW 7.0 wprowadzono nowy rodzaj przyrządu zwany ekspresowym. Po jego umieszczeniu w kodzie natychmiast pojawia się płyta czołowa umożliwiająca konfigurację.

Interakcja pomiędzy płytą czołową i oknem diagramu ma charakter zdarzeń jednocześnie zachodzących na obu obiektach. Przykładem może być umieszczenie kontrolki na płycie czołowej. Operacji odpowiada pojawienie się w oknie kodu zmiennej związanej z tą kontrolką. Należy zauważyć, że kontrolki związane są z **wejściami** obiektów programowych, podczas gdy wyświetlacze odpowiadają **wyjściom** tych obiektów. Każdy obiekt w kodzie posiada zestaw wejść/wyjść zwanych **terminalami**. Odpowiadają one argumentom oraz wartościom zwracanym funkcji w klasycznych językach programowania. Oczywiście jest, że możliwe jest łączenie wyłącznie wejść z wyjściami, każda inna kombinacja wywołuje sprzeciw kompilatora. Łączenie obiektów programowych odbywa się przy pomocy narzędzia połączeniowego (po jego wybraniu kursor zmienia się kołowrotek). Po jego wybraniu należy najechać na obiekt, którego wyjście chcemy połączyć. Wówczas pokazane zostaną wszystkie terminale, rola użytkownika sprowadza się zaś do wybrania konkretnego. Następnie należy przemieścić kursor (od tej chwili będzie się za nim „ciągnąć” przerywana linia – „nitka”) nad obiekt, którego wejście chcemy połączyć z uprzednio wybranym wyjściem. Analogicznie trzeba dokonać wyboru odpowiedniego terminala. Jeśli wszystko zostało wykonane prawidłowo, w oknie kodu powinna pokazać się kolorowa linia łącząca obiekty. Jej kolor odpowiada konkretnemu typowi danych. W przypadku popełnienia błędu, np. połączenia dwóch terminali tego samego typu, pojawi się przerywana linia przekreślona czerwonym krzyżykiem. Oznacza to, że operacja została wykonana niewłaściwie a połączenie należy poprawić.

5.2.5. Konfigurowanie obiektów na panelu czołowym

Umieszczenie kontrolki lub wskaźnika na płycie czołowej przyrządu rzadko wyczerpuje oczekiwania projektanta. bardzo często zachodzi potrzeba np. zmiany zakresu liczb, które można wstawić do kontrolki, zmiany typu danych, umieszczenia w kodzie właściwości obiektu itp. Wszystkie operacje dokonuje się w menu kontekstowym, które uaktywniane jest przez prawy klawisz myszy naciśnięty nad elementem panelu. Poniżej przedstawione są najważniejsze pola tego menu:

- Visible Items – pozwala na skonfigurowanie, które elementy pomocnicze obiektu powinny być widoczne. Np. w przypadku listy zaznaczenie opcji **Digital Display** powoduje umieszczenie obok wybieranej nazwy również liczby jej odpowiadającej.
- Find Terminal – ułatwia znalezienie na diagramie zmiennej odpowiadającej kontrolce. To samo można osiągnąć klikając na niej dwukrotnie lewym przyciskiem myszy.
- Change to Indicator/ Change to Control – zmienia typ obiektu z wyświetlacza na kontrolkę i vice versa.
- Create – jest to podmenu, które umożliwia stworzenie kilku dodatkowych elementów związanych z kontrolką lub wyświetlaczem. Może to być np. zmienna lokalna (**Local**

Variable), stanowiąca po prostu kopię zmiennej, która może zostać wykorzystana w innym miejscu kodu (jest to przydatne, gdy do tego samego obiektu z płyty czołowej trzeba się odwoływać z zupełnie różnych obszarów diagramu). Z drugiej strony menu to umożliwia tworzenie i manipulowanie właściwościami obiektów, takimi jak widoczność, czy blokowanie dostępu (**Visible, Disabled**). Działan tych dokonuje się przy pomocy polecenia **Property Node**.

- Representation – służy do zmiany typu danych reprezentowanego przez obiekt.
- Data Range – umożliwia ustalenie, jakie mogą być wartości zmiennej, zarówno krańcowe (minimalna i maksymalna), jak i początkowa (domyślna).
- Format & Precision – umożliwia zmianę formatu wyświetlanych liczb (automatyczny, naukowy itp.), a także liczbę miejsc po przecinku.
- Data Operations – proste podmenu związane z działaniami na wartości aktualnie reprezentowanej przez zmienną. Może to być np. skopiowanie jej do schowka (**Copy**), lub uczynienie domyślną dla tej zmiennej (**Make Current Value Default**).

5.3. PRZEBIEG ĆWICZENIA

5.3.1. Utworzenie prostego programu do generowania liczb losowych

a) Utworzenie interfejsu użytkownika.

- Uruchomić program „National Instrument LabVIEW 8.5” wybrać opcję „Evaluate”. Następnie w oknie „Getting Started” wybrać „Blank VI”.
 - Co się pojawiło na ekranie?
- W oknie „Untitled1 Front Panel” wstawić przycisk „STOP” (lewy klawisz myszy → Buttons&Switches → Stop Button). Kursorem najechać na wstawiony obiekt, kliknąć prawym przyciskiem myszy i z menu obiektu wybrać "Visible Items". Sprawdzić działanie poleceń "Label", "Caption", "Boolean Text".
 - Jakie są efekty?
- Z palet „Controls” wstawić obiekt „Waveform Chart”, (jw. tj. Express → Graph Indicators → Waveform Chart). Zmienić nazwę „Waveform Chart” na „Liczby Losowe” (double click na tekst nazwy). Zmienić skalę na osi Y na 0-1 (lewy klawisz myszy na oknie Liczby Losowe → Y scale → Formatting → Scales). – **Rysunek 6.**



Rys. 6.

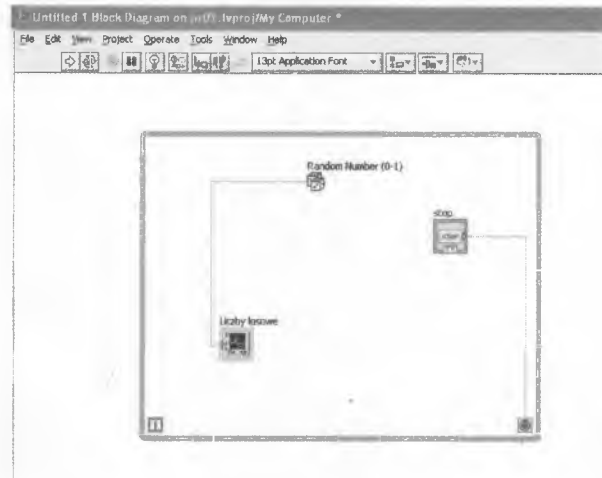
b) Utworzenie i uruchomienie programu w języku graficznym.

- Przejsć do okna „Block Diagram”. Obiekty „Stop” i „Liczby Losowe” są odpowiednikami obiektów wstawionych w oknie „Front Panel”.
- Z palety „Functions” wstawić obiekt „Rundom Number (0-1)” (lewy klawisz myszy → Functions → Mathematics → Numeric → Random Number (0-1)). Najechać kursorem na obiekt „Rundom Number (0-1)” i połączyć linią z „Liczby Losowe”.
- Wstawić pętlę „While Loop” (Functions → Programming → Structures → While Loop).

Wybrany obiekt przenieść do okna "Block Diagram". Nacisnąć i zwolnić lewy przycisk myszy. Przesuwając kursor myszy objąć powstałym prostokątem obiekty, które będą wykonywane w pętli ("Random Number 0-1", "Liczby Losowe", "Stop"). Obiekt „Loop Condition” połączyć z przyciskiem „Stop”. Sprawdzić czy aktywna jest opcja „Stop if True” (lewy klawisz myszy na Loop Condition). Pętla "While Loop" jest wykonywana gdy na wejście przerywające wykonywanie pętli jest podana wartość logiczna "TRUE". Obiekt "Stop" generuje wartość "TRUE" po jego uaktywnieniu. Dlatego należy sprawdzić ustawienie pętli "Continue if True" // "Stop if True".

o✓ Jak LabView sygnalizuje zmianę opcji "Continue if True" na "Stop if True"?

- Uzyskane efekty porównać z rysunkiem 7.

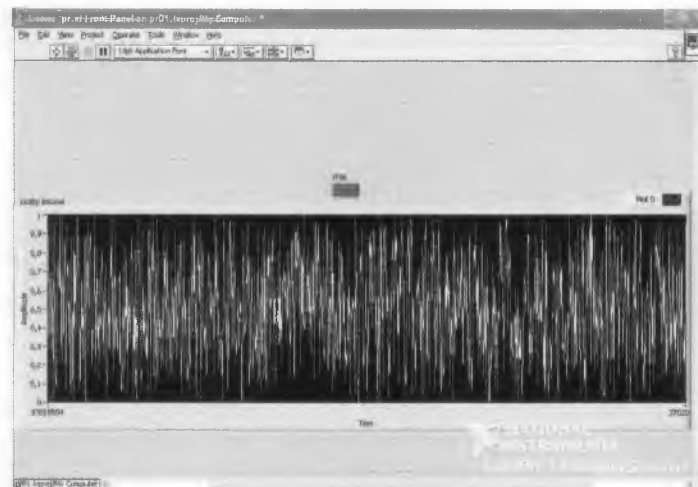


Rys. 7.

- Zapisać program pod nazwą „losowe.vi”.
- Przejść do okna „losowe.vi Front Panel”. Wcisnąć przycisk „Run” a następnie „Stop”.

Rysunek 8.

o✓ Jaki jest efekt działania programu?



Rys. 8.

c) Dodanie taktowania do programu „losowe.vi”.

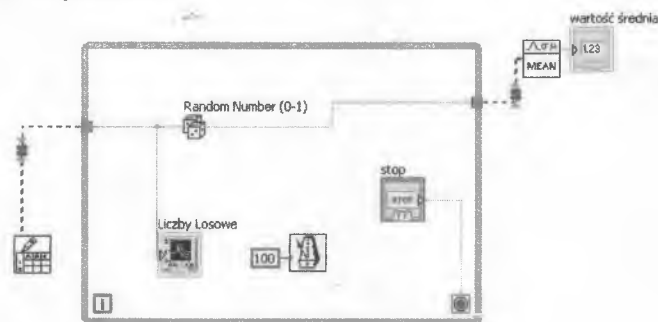
- Wrócić do okna „losowe.vi Block Diagram”. Wstawić wewnątrz pętli obiekt „Wait Until Next ms Multiple” (Functions → Programming → Timing → Wait Until Next ms Multiple). Dodać obiekt „Numeric Constans” (Functions → Mathematics → Numeric → Numeric Constans). Wpisać w oknie „Numeric Constans” wartość 100. Połączyć z obiektem „Wait Until Next ms Multiple”.
- Zapisać zmiany wprowadzone do programu w pliku „losowe2.vi” (Save As → Rename).

Uruchomić program i sprawdzić jego działanie.

- ✓ ○ Co się zmieniło?
- Jak będzie działał program jeśli wpiszę inną wartość w oknie „Numeric Constans”?
- Wrócić do okna „Block Diagram”. Uruchomić program („Run”), uaktywnić przycisk „Highlight Execution/DoNot Highlight Execution”.
- ✓ ○ Jaki jest teraz efekt działania programu?

d) Analiza wyników, zapis w pliku, testowanie programu.

- Z palety "Functions" wybrać obiekt „Mean.vi” (Mathematics → Probability&Statistic → Mean.vi) i umieścić na zewnątrz pętli "While Loop" (uwaga obiekt "Mean.vi" został umieszczony na zewnątrz pętli w celu obliczenie wartości średniej po przerwaniu pętli).
- Wstawić obiekt „Write To Spreadsheet File.vi” (Functions → Programming → File I/O → Write To Spreadsheet File.vi) i umieścić również na zewnątrz pętli "While Loop".
- Kursorem najechać na prawy górny róg obiektu „Mean.vi” i kliknąć prawym przyciskiem myszy. Z menu obiektu wybrać „Create → Indicator”. Zarówno w oknie „Block Diagram” jak i w oknie „Front Panel” pojawi się obiekt „mean” służący do wyświetlania wyników obliczeń, który automatycznie zostanie połączony z obiektem „mean.vi”. Zmienić nazwę obiektu „mean” na „Wartość średnia”.
- Połączyć wyjście obiektu „Random Number (0-1)” z wejściem „X” obiektu „Mean.vi”. Połączenie zostanie narysowane przerywaną linią z czerwonym przekreśleniem co sygnalizuje, że zostały połączone piny reprezentujące różne typy danych.
- W podobny sposób połączyć wyjście „Random Number (0-1)” z wejściem „1D data” obiektu „Write To Spreadsheet File.vi”. Należy wybrać wejście „1D data” ponieważ pętla „While Loop” tworzy jednowymiarową tablicę danych z liczb losowych generowanych przez obiekt Random Number (0-1)”. Połączenie zostanie również narysowane przerywaną linią z czerwonym przekreśleniem, co sygnalizuje, że zostały połączone piny reprezentujące różne typy danych – **rysunek 9**.



Rys. 9.

- Pomarańczowe kwadraty w prostokącie symbolizującym pętlę „While Loop” reprezentują dane generowane przez pętlę, które są przekazywane do dalszej części programu po przerwaniu pętli. Kursorem najechać na pomarańczowy kwadrat i kliknąć prawym przyciskiem myszy. Z menu wybrać opcję „Enable Indexing”. Zmiana tej opcji spowoduje, że pętla będzie gromadziła dane w postaci tablicy jednowymiarowej.
 - (Jak program sygnalizuje zmianę tej opcji?) ✓
- Tablica jednowymiarowa może być wprowadzona na połączone w omawianym przykładzie wejścia obiektów „Mean.vi” i „Write to Spreadsheet File”. Linia przerywana łącząca obiekty zmieni się na pomarańczową linią ciągłą ponieważ niezgodność typów danych została usunięta.
- Zapisać zmiany wprowadzone do programu w pliku „losowe3.vi”.
- Uruchomić program i sprawdzić jego działanie. Po naciśnięciu przycisku „stop” pojawi się

okno dialogowe, w którym należy wpisać nazwę pliku do którego zostaną zapisane wyniki.

5.4. OPRACOWANIE SPRAWOZDANIA

Sprawozdanie powinno zawierać:

- odpowiedź na pytania zadane w części praktycznej ćwiczenia;
- plik (w postaci wydruku) z wygenerowanymi min. 20 liczbami losowymi z przedziału 0-1 w formacie *.txt lub *.xls oraz odpowiadającą im wartość średnią;
- opis czynności umożliwiający wygenerowanie liczb losowych z przedziału, (np. 0-10, lub w przedziale 5-6 itp.) o szczegóły przedziału zapytaj prowadzącego zajęcia;
- plik (w postaci wydruku) z wygenerowanymi min. 20 liczbami w zadanym przedziale w formacie *.txt lub *.xls oraz odpowiadającą im wartość średnią.